

CLAIMS

What is claimed is:

1. An application programming interface, comprising:
 - an input component that receives data and/or instructions; and
 - a meta-model object that is processed by the input component, the meta-data model interrelates data *via* a class hierarchy.
2. The application programming interface of claim 1, the meta-model includes relationship descriptions between class objects.
3. The application programming interface of claim 2, the class objects are associated with a meta-class.
4. The application programming interface of claim 1, the input component is associated with services that operate in conjunction with an operating system framework.
5. The application programming interface of claim 4, further comprising an extension to the framework to interact with at least one of software modeling tools, editor tools, design tools, graphics tools, and word processors.
6. The application programming interface of claim 4, further comprising one or more assemblies that contain annotated classes that derive from classes defined by the framework.
7. The application programming interface of claim 4, the framework reads meta-data from one or more component assemblies to determine at runtime a structure of an item.

8. The application programming interface of claim 7, the item is a document.
9. The application programming interface of claim 1, the meta-object model includes at least one of a meta-class, a meta-attribute, a meta-relationship, an integrity rule, and a behavior description.
10. The application programming interface of claim 9, the meta-class inherits from one or more other meta-classes and/or from base functionality provided by an operating system framework.
11. The application programming interface of claim 9, the meta-class is associated with a relational database.
12. The application programming interface of claim 9, the meta-class is identified *via* a globally unique identifier.
13. The application programming interface of claim 9, the meta-class is identified *via* at least one of a name and a caption to facilitate application development.
14. The application programming interface of claim 1, the meta-model object is associated with at least one of a meta-meta-model and a data model.
15. The application programming interface of claim 9, the meta-class is a class encapsulating data employed to represent another class.

16. The application programming interface of claim 9, the meta-relationship include relationships between model classes.
17. The application programming interface of claim 9, the meta-relation-ship is optionally captured in a database *via* a join operation that allows cardinality combinations to be modeled uniformly.
18. The application programming interface of claim 9, the meta-relationship optionally includes at least one meta-role.
19. The application programming interface of claim 1, further comprising a component to manage at least one of a system state, an event, a transaction, a rollback, and a schema.
20. The application programming interface of claim 19, the transaction is a nested transaction.
21. The application programming interface of claim 19, the transaction is associated with at least one of an undo and a redo operation.
22. A data management system, comprising:
 - a processing component that receives an item that includes meta-data annotations;
 - and
 - an analysis component that determines at runtime a structure of an item *via* deployment of the meta-data annotations.
23. The system of claim 22, further comprising a framework component that defines meta-data class derivations.

24. The system of claim 23, the framework component defines at least one of a meta-data class, a meta-data rule, and a meta-data class behavior.
25. The system of claim 23, the framework further comprises a meta-model object having at least one of a store and a substore to facilitate operations with meta-data components.
26. The system of claim 25, the substore including at least one of an element, an element link, a property, a class field, a model field, a relationship field, a role field, an attribute field, and a schema field to facilitate operations with meta-data components.
27. The system of claim 25, the store is associated with at least one of a cache manager, an element class factory, a model event coordinator, a transaction manager, an undo manager, and a working proxy store.
28. A computer readable medium having computer readable instructions stored thereon for implementing the processing component and the analysis component of claim 22.
29. A method to process meta-data, comprising:
 - determining parameters in accordance with a meta-model class;
 - determining relationships in view of the meta-model class; and
 - inheriting properties to the meta-model class from at least one of another class and an operating system framework to facilitate development of meta-data applications.
30. The method of claim 29, further comprising automatically processing at least one of an event and a transaction when interacting with the meta-model class.

31. The method of claim 29, further comprising providing at least one application programming interface to interact with the meta-model class.
32. The method of claim 29, further comprising processing at least one domain when interacting with the meta-model class.
33. The method of claim 29, further comprising caching a portion of the meta-class from at least one of a client and a server system.
34. The method of claim 33, further comprising presenting the portion of the meta-class to an application as if the portion resided on the client and server system.
35. A system to facilitate meta-data interactions with an operating system, comprising:
 - means for modeling a meta-data object;
 - means for relating the meta-data object to at least one class; and
 - means for interacting with an operating system *via* the meta-data object.
36. A computer readable medium having a data structure stored thereon, comprising:
 - a first data field related to a store associated with a meta-model object; and
 - a second data field related to a substore that describes a portion of the meta-model object.
37. The computer readable medium of claim 36, the substore includes at least one of a meta-model and a data model.
38. The computer readable medium of claim 36, further comprising a data field describing at least one of an element and an element link.

39. The computer readable medium of claim 36, further comprising a data field describing at least one of meta-model information, meta-class information, meta-relationship information, meta-role information, meta-attribute information, domain information, and property information.

40. The computer readable medium of claim 36, further comprising a data field describing at least one of a schema, a table, and a column.

41. The computer readable medium of claim 40, at least one of the schema, table and column is stored on a relational database.